



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

Am

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/696,399	10/25/2000	Michel Casabona	DE9-1999-0075US1	3668
7590	06/06/2005		EXAMINER	
DUKE W. YEE CARSTENS, YEE AND CAHOON, L.L.P. P.O. BOX 802334 DALLAS, TX 75380			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 06/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	09/696,399	CASABONA ET AL.
	Examiner Tuan A. Vu	Art Unit 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM  
THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 28 February 2005.  
 2a) This action is FINAL.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-16 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-16 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____

## DETAILED ACTION

1. This action is responsive to the Applicant's response filed 2/28/2005.

As indicated in Applicant's response, claims 1, 4, 7, 12 have been amended. Claims 1-16 are pending in the office action.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacInnis, USPN: 6,487,723 (hereinafter MacInnis), in view of Saether et al., USPN: 6,405,219 (hereinafter Saether).

**As per claim 1,** MacInnis discloses a method for maintaining software products implemented in a plurality of files in client computer systems (e.g. Fig. 2) located decentralized relative to at least one central software (e.g. *system 200* – Fig. 2) maintenance institution wherein the client systems are connected with said maintenance institution via a network, such method comprising:

providing product information in the network system for making the product information available for said client systems (e.g. *Table T, broadcasts all versions* – col. 4, lines 23-44; Fig. 2); and

performing a software maintenance action for the product (e.g. Fig. 3, col. 5, lines 11-25) from the client site by downloading the data required for said maintenance from a combination

of a top-level repository storing a set of files for the product (e.g. *system* 200, Fig. 2) and a local-level repository storing a first subset of files for the product (e.g. terminal 203, 204, Fig. 2; *internal table* - col. 4, line 45 to col. 5, line 25 – Note: internal table and local storage of files or hardware modules at terminal reads on local-level repository storing subset files specific to a given client), wherein the first subset of files is specific for a given client system.

As for the limitation that the data downloaded from the top-level repository is different from data downloaded from the local-level repository, the data loaded down from both the top-level and local-level repositories is used by the given client in performing the software maintenance action, MacInnis discloses two different set of files, one at a top-level and one at the local level being used in the comparison process at the recipient terminal ( see Fig. 3; *internal table, Table T* - col. 4, lines 22-44), i.e. the internal table being the result of an previous download and the actually downloaded descriptor table T reads on both set of files being downloaded to the terminal and being used therein for comparison purposes. In case the loading down from the local-level repository is not clearly a download, this downloading from a local-level directory as from an intermediate repository before reaching the client target would have been obvious in light of the possibility of establishing intermediate repository as set forth below.

MacInnis does not specify that downloading from a combination of source and local repositories is downloading from a sequence of repositories, even though MacInnis' downloading process is the result of sequential actions taken by a higher-level repository and a local repository. The implementing of repositories in a plurality so to alleviate overburdening storage by one repository and to impart differentiation in purposes to each storage location was a well-known concept at the time of the invention. For example, MacInnis discloses selective

downloading from different sources, e.g. channels or locations or network subsystems, based on descriptor table information provided by a network and retrieving of said versions therefrom (e.g. *loc 320, Fig. 3; network subsystems* - col. 5, lines 21-25; Fig. 4; col. 7, lines 32-58; *combination of channels* - col. 8, lines 41-59). The diversifying of network places, channels or sites for storing and distributing software as suggested by MacInnis (e.g. col. 6, lines 8-15) according to a given specificity or customization criteria is also taught in Saether's following method. Saether, in a method to distribute versions of software files to a network of target machines (e.g. content servers) for maintenance purposes using a distribution list analogous to the MacInnis's descriptor table as shown above, discloses the distribution of software in sequence from a more global server (e.g. *source servers* - Fig. 1) to more secondary global servers before updating the target machines (e.g. Figs. 2-3, 5), each level of server storing files based on a difference identification between server level files (col. 6, line 60 to col. 7, line 32). Based on teachings by Saether as to enable one level of repository to keep a difference of files as compared to another level of repository and the need to store specific set of files at various channels by MacInnis, the local level and top level repository in the sequence of repositories as disclosed by Saether would have been obvious. It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the plurality of storage channels as mentioned by MacInnis into a sequence of repositories going from a high to low level of order as suggested by Saether, because this would alleviate network bandwidth overuse (col. 8, lines 56-67), enable management of network data stored and identification for change at the highest level with the assistance of the intermediate level storage of files to detect file differential - to

obviate redundant replication, to handle the version adjustment and change activation prior to the delivery to the target machine as suggested by Saether ( see col. 5, lines 9-64).

**As per claim 2,** MacInnis discloses a set of repositories, channels or sites for customizing the download of software according to a mapping/comparing process for a selected version or fix for a obsolete version (e.g. Fig. 2-5) but does not expressly disclose a mid-level repository storing a subset of files. Saether teaches an intermediate repository replicating the request of a version propagated from a high level server and build a final set of files for the target download (e.g. primary and secondary global servers - Figs. 2-3, 5) and storing at a lower level only the set of files identified as different from a higher level (col. 6, line 60 to col. 7, line 32), with the set of files to fulfill a version fix. In view of such concept and the teachings by MacInnis to use more than one sites to download the correctly identified version of files, it would have been obvious for one of ordinary skill in the art at the time the invention was made to use mid-level additional sites or repositories as suggested by Saether as an intermediate step for retrieving the correct upgrade files or version files so to enable upgrade preparation, to obviate redundant storage and prevent potential bandwidth overburdening as set forth above in claim 1.

**As per claim 3,** MacInnis does not disclose a fall back to an older program version by inactivating the newer version and activating the older version but teaches seeking of the most suitable version for an operating system ( Fig. 4). The fall back to a previous version upon determining that a new upgrade is not compatible with the target operating system was a known concept and such is evidenced by Saether ( e.g. step 174 -- Fig. 4). It would have been obvious for one of ordinary skill in the art at the time the invention was made to include the rollback step as suggested by Saether to the activation process by MacInnis because this would immediately

and easily restore the failing system, should it encounters problems in activating the upgrade software file, to its functional state without extraneous clean-up operations or costly operating system complications by reactivating the original backup copy with its inherent machine state according to well-known practices.

**As per claim 4**, MacInnis discloses a method for maintaining software products implemented in a plurality of files in client computer systems (e.g. Fig. 2) located decentralized relative to at least one central software ( e.g. *system 200* – Fig. 2) maintenance institution wherein the client systems are connected with said maintenance institution via a network, such method comprising:

providing product information in the network system for making the product information available for said client systems (e.g. *Table T, broadcasts all versions* – col. 4, lines 23-44; Fig. 2); and

performing a software maintenance action for the product (e.g. Fig. 3, col. 5, lines 11-25) from the client site by downloading the data required for said maintenance from a combination of a top-level repository storing a set of files for the product (e.g. *system 200*, Fig. 2) and a local-level repository storing a first subset of files for the product (e.g. terminal 203, 204, Fig. 2; *internal table* - col. 4, line 45 to col. 5, line 25 – Note: internal table and local storage of files or hardware modules at terminal reads on local-level repository storing subset files specific to a given client), wherein the first subset of files is specific for a given client system;

generating of an input list downloadable from a server repository (e.g. *Table T, broadcasts all versions* – col. 4, lines 23-44; Fig. 2);

generating a list of files present on the target client system and comparing of those lists

(e.g. Fig. 2; *internal table* - col. 4, line 45 to col. 5, line 25; Fig. 3A-B);

comparing the list of downloadable files with the list of files present in the target system

(e.g. Fig. 4,5; col. 5, line 11 to col. 6, line 39); and

downloading a plurality of files which are not yet present in the target system (e.g. Fig. 4,5; col. 6, line 40 to col. 7, line 24 ).

As for the limitation that the data downloaded from the top-level repository is different from data downloaded from the local-level repository, the data loaded down from both the top-level and local-level repositories is used by the given client in performing the software maintenance action, MacInnis discloses two different set of files, one at a top-level and one at the local level being used in the comparison process at the recipient terminal ( see Fig. 3; *internal table, Table T* - col. 4, lines 22-44), i.e. the internal table being the result of an previous download and the actually downloaded descriptor table T reads on both set of files being downloaded to the terminal and being used therein for comparison purposes. In case the loading down from the local-level repository is not clearly a download, this limitation would have been obvious in light of the possibility of establishing intermediate repository as set forth below.

MacInnis fails to specify that the downloadable input list is retrieved from a sequence of repositories. But in view of the combined teachings by MacInnis and Saether in addressing the use of a sequence of source and global servers to enhance the step preparation of the upgrade files and alleviate overuse of network or storage resources as set forth in claim 1, this limitation herein would have been obvious for the same rationale as set forth therein.

**As per claim 5**, MacInnis does not specify a total list being a merge of input lists from each repository with a priority of more local files; but discloses a version differential matching of input lists (e.g. Fig. 3-5) with a priority of local files (e.g. *internal table* - col. 4, line 45 to col. 5, line 25). But Saether, in the method of synchronizing of server files using multiple layers of distribution for version and files management ( re claim 1) as mentioned above, discloses the merge into a delivery list of identified files retrieved from with isolated source servers via extracting differential set of files (col. 6, line 60 to col. 7, line 32) to yield a final delivery version list for being activated at the target machines( e.g. Fig. 3A). It would have been obvious for one of ordinary skill in the art at the time the invention was made so that when multiple repositories level are implemented such as suggested by Saether, to modify the use of multi-channel file retrieval and file differential matching as taught by MacInnis and include therein a merging process applied on software list or input lists ( Note: the requirement that priority be given to match local the target machine is inferred or implicitly disclosed from the teachings by Saether with the narrowing of files via merging and differential storing). One of ordinary skill in the art would be motivated to do so because using persistent means for merging files would ensure the non-duplication of unwanted data so well-known in persistence of data and version management processes; and also would lead to a specific and easy-to-propagate set of required files ( e.g. input to a next level of service in sequence) thus enhancing the distribution of tasks imparted at each level of global server; as well as obviate burden in storage and overhead as suggested by Saether (see rationale for using primary and secondary server from claim 1)

**As per claim 6**, MacInnis does not explicitly disclose a look-aside procedure to access in a neighbor system making it easier for integrating the files in the target system but discloses

possible access to the available sites suitable to provide the appropriate component files based on specification of the descriptor table (e.g. Fig. 2-5; *network subsystems* - col. 5, lines 21-25; Fig. 4; col. 7, lines 32-58; *combination of channels* - col. 8, lines 41-59); hence alternate ways to look for a file is suggested. Official notice is taken that a search being performed in a network designed so to provide alternative to reach for the nearest node or target point most easily accessible, or to seek out for the least resistive path was a well-known concept in the search algorithm - notably in network routing and TCP packet delivery - at the time the invention was made. Hence, the look-aside procedure is suggested or implied when multi-channels or network subsites are to be accessed from MacInnis' method or from the delivery list built up in Saether's approach to seek source servers for file gathering from above. Based on this rationale and the above notice, it would have been obvious for one of ordinary skill in the art at the time the invention was made to make sure that the pointing to a external sites as suggested by MacInnis be such that the nearest system be located first as in a look-aside paradigm, like a neighboring system, because this would facilitate the retrieval of files as intended for the upgrade and resources are averted for not having to extend to far reaches or resources straining path to get to repositories for needed files.

**As per claim 7**, MacInnis discloses a system for maintaining software products, comprising:

one central software maintenance site; a network and a plurality of client computer systems decentralized relative to at least one central software ( e.g. Fig. 2) maintenance site wherein the client systems are connected with said maintenance site via a network;

a set of repositories, e.g. a top-level repository for storing a complete set of files for the product and local-level repository for storing a first subset of files, such subset being specific for a given client system (e.g. *system* 200, Fig. 2; terminal 203, 204, Fig. 2; *internal table* - col. 4, line 45 to col. 5, line 25 – Note: re claim 1); to provide product information for a product for making the product information available for said client systems (e.g. Fig. 2; *network subsystems* - col. 5, lines 21-25; col. 7, lines 32-58; *combination of channels* – col. 8, lines 41-59);

wherein a given client computer system performs a software maintenance action for the product by downloading data required for said software maintenance action (e.g. Fig 3-5),

As for the limitation that data downloaded from the top-level repository is different from data downloaded from the local-level repository and the data loaded down from both the top-level and local-level repositories is used by the given client in performing the software maintenance action; this limitation has been addressed in claim 1. In case the loading down from the local-level repository is not clearly a download, this limitation would have been obvious in light of the possibility of establishing intermediate repository as set forth below.

MacInnis does not disclose that the set of repositories is a sequence of repositories nor does MacInnis explicitly teach that downloading from a combination of source and local repositories is downloading from a sequence of repositories. But these limitations have been addressed in claim 1 above.

**As per claim 8**, MacInnis does not disclose a sequence of repositories hierarchically arranged but Saether discloses a system of hierarchically arranged repositories (e.g. Fig. 1, 5-6). This limitation would have been obvious using the same rationale and motivation set forth in

claim 1 above by combining the subsystems and multi-channels file retrieval by MacInnis to the hierarchy of servers as taught by Saether.

**As per claim 9,** this claim corresponds to claim 2, hence is rejected using the same grounds of rejection as set forth therein.

**As per claim 10,** this claim corresponds to claim 6, hence is rejected using the same grounds of rejection as set forth therein.

**As per claim 11,** only Saether discloses replication repositories ( e.g. *2<sup>nd</sup> global servers* – Fig. 1, 3, 4) for receiving input list from a higher level server and re-constructing of the delivery list. It would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the multi-location or channel storage and downloading adjustment at the local level repository to implement replication and intermediate service, i.e. shadow repositories, as suggested by Saether to help recording and duplicating of delivery data for target upgrade activation and information propagating purposes for the same reasons as set forth in claim 1 above.

**As per claim 12,** this claim is the computer readable medium version of method claim 1, hence is rejected using the corresponding rejection as set forth therein.

**As per claim 13,** this claim is the computer readable medium version of method claim 2, hence is rejected using the corresponding rejection as set forth therein.

**As per claim 14,** this is the computer program product version of claim 4 for which MacInnis discloses instructions for performing maintenance action for an upgrade of a program on one target, such action including instructions:

generating (input list downloadable);

generating (list of files present on the target client) and comparing of those lists; comparing (list of downloadable files with the list of files present in the target); and downloading (files which are not yet present); all of which limitations having been correspondingly addressed in claim 4.

But MacInnis fails to specify that the downloadable input list is retrieved from a sequence of repositories. But in view of the combined teachings by MacInnis and Saether in addressing the use of a sequence of service repositories to improve the task/storage repartition and overhead or overuse of bandwidth resource as set forth in claim 1, this limitation herein would have been obvious for the same rationale as set forth therein.

**As per claims 15-16**, these claims are the computer program product versions of method claims 5 and 6, respectively, hence are rejected using the corresponding rejection as set forth therein.

*Response to Arguments*

4. Applicant's arguments filed 2/28/2005 have been fully considered but they are not persuasive. Following are the observations by Examiner in regard thereto.

**Rejections under 35 USC §103(a):**

(A) Applicants have submitted that information in the internal tables is not downloaded (Appl. Rmrks, pg. 8, top para). In response, it is noted that there is a low-level environment at MacInnis terminal where a comparing process takes place; this is a machine level execution to which data (internal table files) are loaded from storage like a file system level ( Note: data in non-volatile memory like disk file would always be there when needed for comparing); and it is considered a form of loading down onto a comparing process execution, with MacInnis' terminal

--where such execution takes place -- being the so-called low-level repository or client machine. The claim recites 'downloaded from a local-level repository'; and as shown in MacInnis, the terminal by storing a internal table file being persisted so as to be loaded to a machine level comparing process can be viewed as downloading a file. For the sake of argument, even if MacInnis has to download files from a mid level repository in order to perform the comparison, the mid level repository limitation would have been obvious in view of the teachings by Saether and the teachings by MacInnis; and this Saether's teaching is the sequence of repositories subject matter being addressed under the obviousness rationale as set forth in claim 1.

(B) Applicants have submitted that Saether method does not maintain software for client systems (Appl. Rmrks, pg. 8, middle para) and that sequences of server distribution does not provide motivation that it would take for managing/accommodating the needs of a client system. The subject matter at stakes is not server or client but narrowing a larger set of downloadable files so that only a particular set thereof is determined based on some comparing schemes so as to only allow such a specific and smaller set of files to be retrieved at the target ends. The rationale as set forth in the rejection has demonstrated how Saether being using comparing obviate storing of duplicate and redundant set of files between Primary and Secondary server ( see col. 6, line 60 to col. 7, line 32). Whereas the purpose is to try to alleviate storage resources, whether it is a server machine or a client machine, the point would be moot; and the rationale in the rejection has set up why Saether's teaching combined with the need by MacInnis would have render obvious the sequence of repositories. In response to applicant's argument that server application and client system are nonanalogous arts, it has been held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the

particular problem with which the applicant was concerned, in order to be relied upon as a basis for rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In this case, it is also noted that it is the apparent purpose which both the claimed invention and the prior art references being combined strive to attain that needs to be addressed, not the field of technology (i.e. server application versus client application) in which such purpose is being obtained.

(C) Applicants have submitted that Saether does not teach 'top level repository storing a set ... local repository storing ... first subset ... for a given client system' (Appl. Rmrks, pg. 9, top para). This argument falls under the conviction that Saether does not provide client upgrade; and that Saether repositories only serve as duplicated storages not fulfilling a set/subset paradigm and accommodating a client needs. These arguments are to be referred back to section B above.

(D) Applicants have submitted that Saether and MacInnis do not teach the limitation of multiple sources as required by claim 4 and that *prima facie* case of obviousness is improper (Appl. Rmrks, pg. 9, bottom, pg. 10, top 2 para). The rejection of claim 4 would have been same as established for claim 1; hence the reply for these arguments is referred back to section A and B above.

(E) Applicants have submitted that the comparing of MacInnis and the merging by Saether do not disclose 'priority of more local files' as required from claim 5 (Appl. Rmrks, pg. 10, bottom para). The recited phrase 'with priority of local files' has been interpreted as emphasis to accommodate the more specific needs of target system; and by merging from higher server and mid-level server as by Saether or by using specifics version differentiation by MacInnis from channel provider, Saether and MacInnis as combined have met the limitation as to accommodate

the merging process so to yield a final set files specific to the need of a target machine. The claim for not being more specific about 'with a priority of local files' will not be treated in any particular way ( like that alleged by Applicants) other than what has been construed by Examiner via broad and reasonable interpretation. The arguments (Appl. Rmrks, pg. 11, top para) against merge and configuring server files versus local client files would have to be referred back to section B above.

(D) Applicants have submitted that Examiner's use of 'well-known' concept is impermissible and that piecing together the claimed invention via utilizing pieces of prior art is inappropriate (Appl. Rmrks, pg. 11, bottom para ). The rejection has invoked a well-known concept in network transmission or data routing to address looking aside for a nearest non-resistive node/path in the network for achieving a data processing or transferring. Applicants have not raised the point whether such known concept is justified or justly corroborated; and instead contend with the assertion that there is no proper *prima facie* without pointing out specifics as to why the rationale used in the combined teachings would be inapposite with the intended purposes of the references. Hence, the rejection still stand for lack of grounds from Applicants identifying the flaw in the combination of teachings as set forth for addressing the look-aside limitation in claim 6.

(E) Applicants have submitted that MacInnis unidirectional transmission does not require look aside procedure (Appl. Rmrks, pg. 12, middle para ). Claim 6 is rejected based on a combination of teachings including the level of repositories by Saether and the look-aside technique so well known in network data routing/transmitting, which teaches seeking a next non-resistive node by a network algorithm to reach a target. In response to applicant's arguments

against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(F) Applicants have submitted that Saether's desire to update content servers does not teach client systems nor does Saether express any desire to look aside ( Appl. Rmrks, bottom pg. 12). The client/server field of application has been addressed in section B; and in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

The sequence of repositories has been viewed as being disclosed by Saether in combination with MacInnnis; and the remaining arguments against the rejection of claims 7, 8-11, 12-16 would have to be referred back to the sections above. The rejection will stand as set forth in the Office Action.

### *Conclusion*

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence – please consult Examiner before using) or 703-872-9306 ( for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

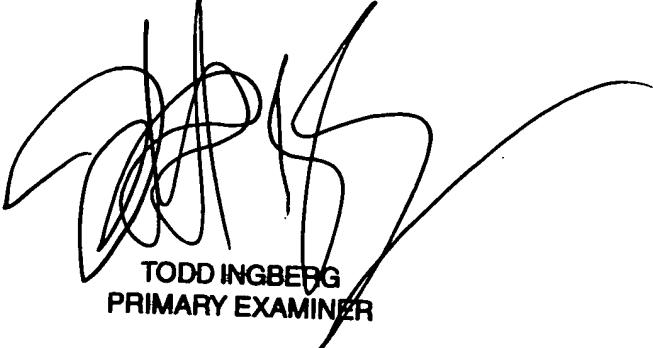
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT

Application/Control Number: 09/696,399  
Art Unit: 2193

Page 18

May 19, 2005



TODD INGBERG  
PRIMARY EXAMINER